What is claimed is:

1.   A method of integrating a first software component with a second software component, the method comprising the steps of:

creating a schema;

integrating the schema into a data wedge;

populating a data model in the data wedge; and

translating data elements in the data model from a first format of the first software component schema to a second format of the second software component.

2.   The method as claimed in claim 1, further comprising the step of:

triggering an event to notify the second software component of translated data element availability.

3.   The method as claimed in claim 1, further comprising the step of:

removing obsolete data elements from the data model.

4.   The method as claimed in claim 1, further comprising the step of:

creating an instance of a data wedge.

5.   The method as claimed in claim 1, wherein the schema created includes a data model and a data wedge name.

6.   The method as claimed in claim 1, wherein integrating the schema into the data wedge includes setting default data elements and data values.

7.   The method as claimed in claim 1, further comprising the step of:

modifying a data element in the data model of the first software component.

8.   A computer system for integrating software components comprising:

a processor; and

a memory coupled to said processor; the memory having stored therein data and sequences of instructions which, when executed by said processor, cause said processor to integrate software components by causing the processor to:

create a schema, integrate the schema into a data wedge, populate a data model in the data wedge, and translate data elements in the data model from a first format of a first software component schema to a second format of a second software component.

9.   The system as claimed in claim 8, further comprising instructions which, when executed by said processor, cause said processor to:

trigger an event to notify the second software component of translated data element availability.

10.   The system as claimed in claim 8, further comprising instructions which, when executed by said processor, cause said processor to:

remove obsolete data elements from the data model.

11.   The system as claimed in claim 8, further comprising instructions which, when executed by said processor, cause said processor to:

create an instance of a data wedge.

12.   The system as claimed in claim 8, further comprising instructions which, when executed by said processor, cause said processor to:

modify a data element in the data model of the first software component.

13.   The system as claimed in claim 8, wherein the schema created includes a data model and a data wedge name.

14.   The method as claimed in claim 8, wherein integrating the schema into the data wedge includes setting default data elements and data values.

15.   A computer implemented system for integrating a first and second software component having a first and second schema respectively and a first and second data view respectively, the system comprising:

a data wedge configured to translate a data element from the first data view in accordance with the first schema to the second data view in accordance with the second schema.

16.   The system as claimed in claim 15, wherein said data wedge is further configured to trigger an event to notify the second software component of translated data element availability.